Unipept in 2024: Expanding Metaproteomics Analysis with Support for Missed Cleavages, Semi-Tryptic and Non-Tryptic Peptides

Tibo Vande Moortele[1], Bram Devlaminck[1], Simon Van de Vyver[1], Tim Van Den Bossche[2,3], Lennart Martens[2,3], Peter Dawyndt[1], Bart Mesuere[1], Pieter Verschaffelt[1,2,3,*]

[1] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, 9000 Ghent, Belgium
[2] VIB - UGent Center for Medical Biotechnology, VIB, 9000 Ghent, Belgium
[3] Department of Biomolecular Medicine, Faculty of Medicine and Health Sciences, Ghent University, 9000 Ghent, Belgium

(*) unipept@ugent.be

Abstract

Unipept, a pioneering software tool in metaproteomics, has significantly advanced the analysis of complex ecosystems by facilitating both taxonomic and functional insights from environmental samples. From the onset, Unipept's capabilities focused on tryptic peptides, utilizing the predictability and consistency of trypsin digestion to efficiently construct a protein reference database. However, the evolving landscape of proteomics and emerging fields like immunopeptidomics necessitate a more versatile approach that extends beyond the analysis of tryptic peptides. In this article, we present a significant update to the underlying index structure of Unipept, which is now powered by a Sparse Suffix Array index. This advancement enables the analysis of semi-tryptic peptides, peptides with missed cleavages, and non-tryptic peptides such as those encountered in other research fields such as immunopeptidomics (e.g. MHC- and HLA-peptides). This new index benefits all tools in the Unipept ecosystem such as the web application, desktop tool, API and command line interface. A benchmark study highlights significantly improved performance in handling missed cleavages, preserving the same level of accuracy.

Keywords

Meta-omics, metaproteomics, bioinformatics, software, data analysis

Introduction

Unipept[1] is an established ecosystem of user-friendly tools for the downstream analysis of metaproteomics data, initially launched as a web application in 2012 and expanded over the years to include a REST API[2], a command line interface (CLI)[2], and a desktop application[3]. All tools in the Unipept ecosystem depend on the same underlying API server, which can be either accessed via our public server or self-hosted for customized workflows.

Unipept excels in rapidly annotating a set of tryptic peptides with taxonomic and functional information. It accomplished this by utilizing a highly optimized relational database. For every tryptic peptide in UniProt[4], Unipept's database stored the precalculated taxonomic lowest common ancestor (LCA)[5] and the linked functional annotations (EC[6], GO[7,8], InterPro[9]). For version 2024_03 of UniProt, this resulted in taxonomic and functional information on 1.3 billion tryptic peptides with a total database size of 1.875 TB. This required a server with at least 60 GB of memory to efficiently retrieve results. For example, it only took 11 seconds to analyze the SIHUMIx[10] sample in the CAMPI benchmark dataset[10] of 24424 tryptic peptides.

First and foremost, in metaproteomics, samples are commonly prepared by adding trypsin to cleave proteins at specific sites[11]. This is why constructing a precomputed index of all unique tryptic peptides from UniProtKB is feasible. From time-to-time, however, trypsin will miss a cleavage site, creating a so-called missed cleavage. By default, prior versions of Unipept

assumed there are no missed cleavages present in a peptide, which means these cannot be matched to proteins. To solve this problem, a solution was retro-fitted to handle these missed cleavages. By enabling the “advanced missed cleavage” setting, Unipept scans the input peptide for missed cleavage positions, splits the peptides and performs separate lookups for each tryptic fragment. The resulting set of proteins are intersected, which results in a collection of possible protein matches, potentially containing the original peptide. Every protein in this collection is then scanned left-to-right to ensure that the original peptide is present, and not only its separate tryptic fragments. All these extra steps have a significant impact on performance. Enabling “advanced missed cleavage handling” increased the analysis time from 11 seconds to 6 minutes 33 seconds for processing the SIHUMIx dataset S07. In addition, the API and CLI did not support missed cleavage handling. Reviewing literature citing Unipept and our web server logs both reveal that almost all users enable missed cleavage handling, underscoring the importance of optimizing its performance. Although this feature significantly slows down the analysis, it has been widely adopted for its ability to enhance analysis accuracy.

Secondly, Unipept could not deal with semi-tryptic peptides (i.e. peptides where only one terminus is tryptic). This problem becomes particularly significant when proteolytic activity is less predictable, such as in the complex environment of the gut microbiome[12].

Finally, support for matching arbitrary peptides such as non-tryptic peptides, which are often encountered in neighboring fields like immunopeptidomics[13], was lacking in Unipept. Immunopeptidomics involves the analysis of HLA (human leukocyte antigen) peptides that play a crucial role in immune system functioning, as they are involved in presenting fragments of proteins (peptides) to T cells, enabling the immune system to distinguish between self and foreign antigens.

In this manuscript, we introduce a significant update to Unipept, consisting of a new index structure based on Sparse Suffix Arrays (SSA). This new index replaces the old relational database with precomputed data. Using this new index, Unipept can match arbitrary peptides (i.e. also semi-tryptic or non-tryptic peptides) instead of only tryptic peptides. In addition, it eliminates the performance penalty of missed cleavage handling. Because this index is used at the API-level of Unipept, these new features are automatically available to all tools in the Unipept ecosystem.

Methods

In the Unipept ecosystem, components such as the Unipept Web application, Unipept Desktop app, CLI and third-party applications submit peptides to a central API for analysis. By making improvements to this API, all tools in the ecosystem automatically benefit. To support arbitrary peptide matching, the relational database in the Unipept API was replaced by a sparse suffix array (SSA), an index designed for efficient searching across multiple sequences. In addition, we completely rewrote the API itself from Ruby-on-Rails to Rust, a low-level, performance oriented programming language which achieves better performance and memory efficiency.

Peptides from a user sample are sent to the API, which in turn processes these peptides by querying the SSA index. This index matches them against all proteins in the UniProtKB database. Relevant taxonomic and functional annotations are retrieved from an optimized in-memory datastore, also implemented in Rust. The API then compiles this information and returns a list of proteins corresponding to each peptide, along with their associated taxonomic and functional annotations. These results are then passed back to the user (see Figure 1). Note that instead of relying on precalculated values, the taxonomic and functional annotations are now calculated on-the-fly.

The SSA index organizes all suffixes of a given set of strings in sorted order, enabling fast retrieval of matching subsequences. Each suffix is stored as a 64 bit integer, pointing to its starting position in the text. Due to the sheer number of suffixes that must be referenced, the SSA index demands substantial memory. Therefore, we employed a number of techniques to reduce memory usage. These include sparseness (retaining only a subset of the suffix array without compromising accuracy), compression and bit packing. The following sections will explore these methods in detail, discussing their effects on both performance and memory efficiency.

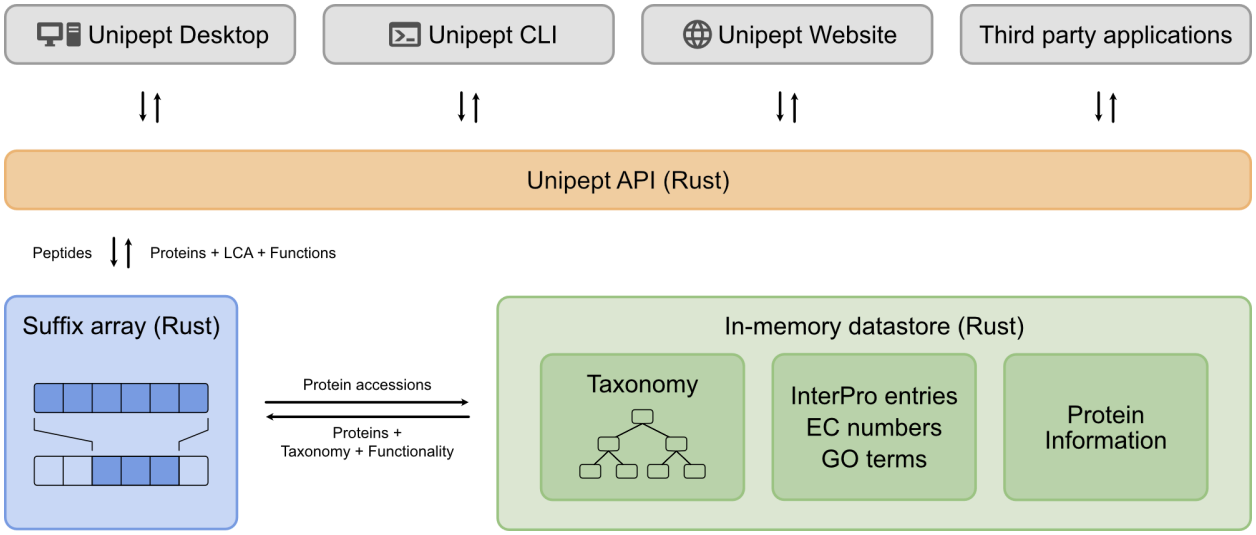


Figure 1: Current architecture of Unipept. Because each component of the Unipept ecosystem (in gray) relies on a common API (in orange) for their data analyses, improving this API immediately enables support for arbitrary peptide analysis and boosts performance for the entire ecosystem.

Constructing the suffix array

The SSA index is built by concatenating all protein sequences in the UniProtKB database. Given the vast number of proteins, this concatenated text and its corresponding suffix array can grow very large. Version 2024_04 of UniProtKB contains nearly 246 million proteins, totalling approximately 87 billion amino acids. Since each suffix is represented by a 64-bit integer, and the number of suffixes matches the number of amino acids, the resulting dense suffix array requires $64 \times 87 \times 10^9$ bits, equivalent to 696 GB of memory.

The suffix array itself can be constructed using the popular libdivsufsort[14] or libsais[15] libraries. Both implementations provide linear-time construction algorithms with a *5n + O(1)* memory complexity, where *n* is the size of the text being indexed. For version 2024_04 of UniProtKB, constructing the suffix array requires a machine with at least 750 GB of memory. Using the libdivsufsort library on a single core of a modern CPU (AMD EPYC 7773X), this process takes approximately 5 hours of computation time.

Sparse suffix array

Although a dense suffix array provides optimal performance, it demands a substantial amount of memory. To decrease the amount of memory required for storing the suffix array, a sampling step can be introduced, resulting in a sparse suffix array (SSA). In an SSA, only every *k*-th suffix of the input text is retained, reducing the array's memory footprint to $1/k^{th}$ of the original.

However, using a sparse suffix array introduces a limitation: an SSA with sparseness *k* only supports searching peptides that are at least *k* amino acids long, and overall lookup performance may decrease. Luckily, this compromise does not introduce a problem for Unipept, since it already restricted the search to peptides with a minimum length of 5 amino acids This aligns with limitations of mass spectrometery which struggles to reliably detect very short peptides. Moreover, such short peptides are prevalent in many proteins and tend to produce overly generic functional and taxonomic analysis results, offering limited information.

By using a sparseness factor of *k = 3*, we significantly reduce the memory requirements of the suffix array while preserving efficient search capabilities. From the original dense suffix array size of 696 GB, this approach reduces the memory usage to 696 / 3 = 232 GB, resulting in a much more manageable sparse suffix array.

Suffix array compression

In addition to increasing the sparseness factor *k,* memory requirements can be further decreased by compressing the (sparse) suffix array. Each suffix represents a position in the text, and while the Rust programming language defaults to using 64 bits per suffix, this is unnecessarily large for the 87 billion amino acids in UniProt. To represent these 87×10^9 different suffixes, we only need $\lceil \log_2(87 * 10^9) \rceil = 37$ bits per suffix, eliminating 27 unused bits per suffix. By storing all suffixes sequentially in a custom bitarray, this compression significantly reduces memory usage.

For a sparse suffix array with *k = 3*, which initially requires 232 GB of memory, applying this compression reduces the size to just 133 GB—a 57% reduction in memory footprint. This optimization allows deploying the suffix array on servers with significantly less memory. Figure 2 illustrates the relative memory usage of each component within the final SSA index.

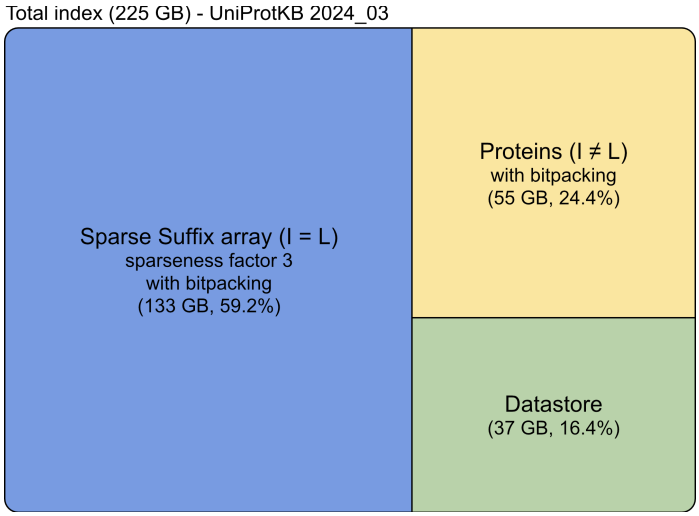


Figure 2: Memory footprint of the individual components in the SSA index built from UniProtKB version 2024_03.

Equating Isoleucine and Leucine

In metaproteomics data analysis, the amino acids isoleucine (I) and leucine (L) are typically treated as equal. To support this requirement, the new index must natively accommodate the equivalence of I and L. Previous versions of Unipept achieved this by duplicating the entire database with I and L equated (I=L), effectively maintaining two versions of the index. However, applying this duplication strategy to the suffix array would double the memory requirements, making it impractical. Therefore, we only construct an index where every occurrence of leucine (L) is replaced with isoleucine (I).

When the “equate I=L” option is enabled, we similarly substitute leucine with isoleucine in the query peptides and directly return the results. For searches where isoleucine and leucine are treated as distinct, an additional filtering step is applied. In this step, each match from the initial search is verified by comparing the occurrences of I and L in the original, unmodified peptide with the corresponding amino acids in the unmodified protein sequences. Any mismatches where I and L were incorrectly equated are filtered out, ensuring only valid matches proceed.

Maximal number of matches

A single peptide can potentially match thousands of proteins. Because we now aggregate all protein matches on demand, this can significantly slow down an analysis. Moreover, the results for such peptides are often quite generic because of their high occurrence. Out of all 1.3 billion tryptic peptides in UniProtKB only 0.001% result in more than 10 000 protein matches. For 95% of these extremely common peptides, the taxonomic lowest common ancestor[16] corresponds to the root node of the NCBI taxonomy (as detailed in the supplementary data S1). Rather than explicitly computing the lowest common ancestor, we opted to heuristically assign the root taxon to all peptides with more than 10 000 matches. This heuristic greatly improves performance

while causing only a minimal reduction in accuracy - in UniProt 2023.3, only 639 out of 1.3 billion tryptic peptides are assigned a more generic LCA under this method.

Results

Replacing the relational database with our novel SSA index should have minimal impact on the accuracy of the analysis results. However, we anticipate a significant performance improvement, particularly when missed-cleavage handling is enabled. To evaluate this impact, we analyzed samples from the highly used SIHUMIx benchmark dataset[10] using the Unipept web application. Version 5.0 of Unipept relies on the old relational database, whereas version 6.0 employs the new SSA index.

Accuracy comparison

To assess the impact on accuracy, we used sample S07 from the SIHUMIx dataset containing 24424 peptides. The full results can be found in supplementary data S2. In the analysis with Unipept 5.0, 297 peptides (1%) could not be matched. In contrast, Unipept 6.0 successfully matched every peptide in the sample. When examining the taxonomic annotations for each peptide, we observed that in Unipept 5.0 2,680 peptides were annotated with an LCA at the species level, whereas this number slightly decreased to 2,551 for Unipept 6.0.

The slight shift in results arises from a fundamental difference in how the relational database and the SSA index handle peptide queries. In the relational database, peptides were assumed to be tryptic, implicitly incorporating information about the amino acids preceding and following the peptide. In contrast, the SSA index can match arbitrary peptides, so this implicit context is no longer considered during matching, which may result in additional protein matches. Given the minimal impact of these differences, we have decided not to implement some kind of tryptic emulation of the old database.

Performance comparison

To evaluate performance, we analyzed six tryptic peptide samples from the SIHUMIx data set. Each sample contained around 25 000 peptides, including both tryptic peptides and peptides with missed cleavages. Duplicate peptides were not filtered and the “equalize IL” option was enabled. Figure 3 shows wall clock times for searching these samples against Unipept 5.0 with and without missed cleavage handling, as well as Unipept 6.0, which natively supports missed cleavage handling.

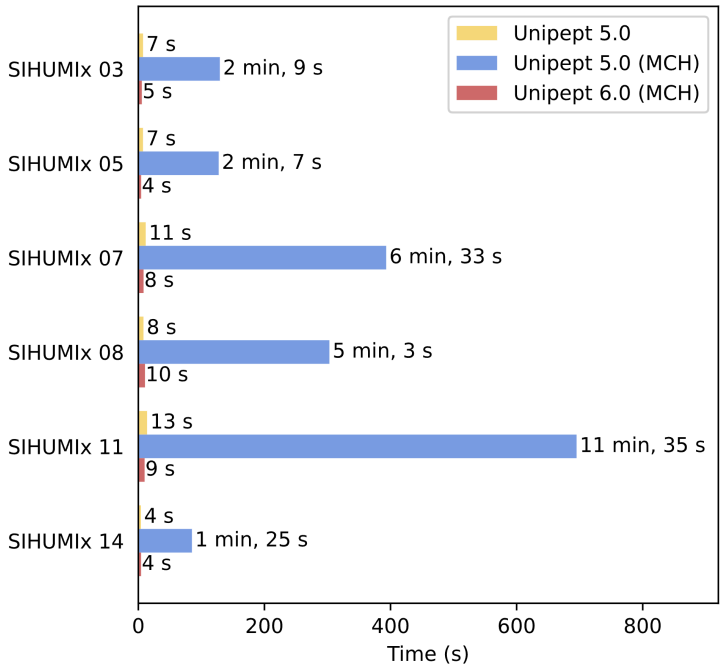


Figure 3: Execution times for Unipept 5.0, and Unipept 5.0 with missed cleavage handling (MCH) enabled and Unipept 6.0. Each SIHUMIx sample contains approximately 25000 tryptic peptides. All analyses used the I = L option.

As a general observation, Unipept 5.0 without missed cleavage handling processes peptides at the same speed as Unipept 6.0 with missed cleavage handling. This represents a 20-70 times speed improvement compared to using missed cleavage handling using Unipept 5.0. For example, when analyzing CAMPI[17] dataset S07, the search time with missed cleavage handling dropped from 6 minutes 33 seconds using Unipept 5.0 to just 8 seconds with Unipept 6.0. This is comparable to the 11 seconds it took to analyze the dataset using Unipept 5.0 without missed cleavage handling.

Comparison with other tools

Tools such as the UniProt peptide search tool[18,19], the Expasy ScanProsite tool[20] and Unipept all provide functionality for matching peptides against the UniProtKB database. In this section, we compare the feature sets and performance of these tools with Unipept 6.0. Because of the poor performance of some of these tools, our tests were restricted to a single, randomly selected tryptic peptide: ISPAVLFVIVILAVLFFISGLLHLLVR.

The features and execution times of all tools are available in Table 1. Both the UniProt peptide search tool and Unipept 6.0 provide similar features, including the ability to use the I=L option and perform searches across the entire UniProtKB database. However, their performance differs significantly: while the UniProt peptide search tool completes searches in a few seconds to minutes, Unipept 6.0 completes the same task in under a few milliseconds.
The Expasy ScanProsite tool adopts a different approach, offering a broad array of approximate matching options. However, its search is limited to a subset of UniProtKB, specifically proteins

from reference genomes, which account for only about one-third of the entire UniProtKB database. Searching for the reference peptide with ScanProsite took 5 minutes.
Unipept 5.0 and Unipept 6.0 both support searching the entire UniProtKB database and allow the I=L option, performing equivalently in terms of accuracy. However, Unipept 5.0 is restricted to tryptic peptides, whereas Unipept 6.0 does not have this limitation.

	UPS	ESP	Unipept 5.0	Unipept 6.0
Used proteins	all	Ref. proteomes	all	all
Approximate match	[IL]	flexible	[IL]	[IL]
Execution time	1 s - 20 min	5 min	**< 5 ms**	**< 5 ms**
Searchable peptides	all	all	~ tryptic	all

Table 1: Comparison of Unipept 5.0, Unipept 6.0, UniProt peptide search (UPS) tool and ExpasyScanProsite (EPS) tool. [IL] in the approximate matching row indicates only I and L can be equated, while ~ tryptic in the searchable peptides row indicates only tryptic peptides or tryptic peptides with missed cleavages can be found.

Conclusions

In this manuscript, we present a significant update to the underlying index structure of Unipept. This index is implemented in Rust and allows fast on-demand calculation of both taxonomic and functional annotations for a set of peptides. Because data processing no longer relies on precalculated results for tryptic peptides, Unipept can now match arbitrary peptides. This allows us to perform “missed cleavage handling” by default up to 70 times faster, making it feasible to integrate missed cleavage handling into the API, CLI, and third-party tools—a capability that was previously unavailable.

Looking ahead, Unipept is well-positioned to continue evolving, with a robust foundation for migrating features currently exclusive to the desktop application onto the web platform. One prominent example is the potential introduction of an ad-hoc filtering feature for UniProtKB, similar to the custom database construction already available in the Unipept Desktop app. This functionality would allow researchers to query a targeted subset of proteins within the extensive UniProtKB database, leveraging taxonomic data such as results from prior metagenomic experiments. By transitioning these advanced features to the web, Unipept can enhance accessibility for a wider range of users, eliminate the need for local software installations, and improve the overall user experience.

Acknowledgements

PV acknowledges funding by Ghent University [BOF/01P10623]. L.M. acknowledges funding from the Research Foundation Flanders (FWO) [G028821N] and [G010023N]. TVDB acknowledges funding by the Research Foundation Flanders (FWO) [1286824N].

Data Availability Statement

Unipept is available as open-source software, with its source code accessible on GitHub. The SIHUMI samples analyzed in this study are publicly available through the CAMPI dataset. Additional data files related to this study are provided as supplementary data.

Supporting information

The following supporting information is available free of charge at [ACS website]:

- Supplementary Data S1. In-depth analysis of tryptic peptides and associated UniProtKB proteins
- Supplementary Data S2. Data validation files

References

(1) Gurdeep Singh, R.; Tanca, A.; Palomba, A.; Van der Jeugt, F.; Verschaffelt, P.; Uzzau, S.; Martens, L.; Dawyndt, P.; Mesuere, B. Unipept 4.0: Functional Analysis of Metaproteome Data. *J. Proteome Res.* **2019**, *18* (2), 606–615. https://doi.org/10.1021/acs.jproteome.8b00716.

(2) Verschaffelt, P.; Van Thienen, P.; Van Den Bossche, T.; Van der Jeugt, F.; De Tender, C.; Martens, L.; Dawyndt, P.; Mesuere, B. Unipept CLI 2.0: Adding Support for Visualizations and Functional Annotations. *Bioinformatics* **2020**, *36* (14), 4220–4221. https://doi.org/10.1093/bioinformatics/btaa553.

(3) *Unipept Desktop 2.0: Construction of Targeted Reference Protein Databases for Metaproteogenomics Analyses | Journal of Proteome Research*. https://pubs.acs.org/doi/full/10.1021/acs.jproteome.3c00091 (accessed 2024-09-23).

(4) The UniProt Consortium. UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **2023**, *51* (D1), D523–D531. https://doi.org/10.1093/nar/gkac1052.

(5) Van Den Bossche, T.; Verschaffelt, P.; Vande Moortele, T.; Dawyndt, P.; Martens, L.; Mesuere, B. Biodiversity Analysis of Metaproteomics Samples with Unipept: A Comprehensive Tutorial. In *Protein Bioinformatics*; Lisacek, F., Ed.; Springer US: New York, NY, 2024; pp 183–215. https://doi.org/10.1007/978-1-0716-4007-4_11.

(6) Webb, E. Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.; 1992.

(7) Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; Harris, M. A.; Hill, D. P.; Issel-Tarver, L.; Kasarskis, A.; Lewis, S.; Matese, J. C.; Richardson, J. E.; Ringwald, M.; Rubin, G. M.; Sherlock, G. Gene Ontology: Tool for the Unification of Biology. *Nat. Genet.* **2000**, *25* (1), 25–29. https://doi.org/10.1038/75556.

(8) The Gene Ontology Consortium; Aleksander, S. A.; Balhoff, J.; Carbon, S.; Cherry, J. M.; Drabkin, H. J.; Ebert, D.; Feuermann, M.; Gaudet, P.; Harris, N. L.; Hill, D. P.; Lee, R.; Mi, H.; Moxon, S.; Mungall, C. J.; Muruganugan, A.; Mushayahama, T.; Sternberg, P. W.; Thomas, P. D.; Van Auken, K.; Ramsey, J.; Siegele, D. A.; Chisholm, R. L.; Fey, P.; Aspromonte, M. C.; Nugnes, M. V.; Quaglia, F.; Tosatto, S.; Giglio, M.; Nadendla, S.; Antonazzo, G.; Attrill, H.; dos Santos, G.; Marygold, S.; Strelets, V.; Tabone, C. J.; Thurmond, J.; Zhou, P.; Ahmed, S. H.; Asanitthong, P.; Luna Buitrago, D.; Erdol, M. N.;

Gage, M. C.; Ali Kadhum, M.; Li, K. Y. C.; Long, M.; Michalak, A.; Pesala, A.; Pritazahra, A.; Saverimuttu, S. C. C.; Su, R.; Thurlow, K. E.; Lovering, R. C.; Logie, C.; Oliferenko, S.; Blake, J.; Christie, K.; Corbani, L.; Dolan, M. E.; Drabkin, H. J.; Hill, D. P.; Ni, L.; Sitnikov, D.; Smith, C.; Cuzick, A.; Seager, J.; Cooper, L.; Elser, J.; Jaiswal, P.; Gupta, P.; Jaiswal, P.; Naithani, S.; Lera-Ramirez, M.; Rutherford, K.; Wood, V.; De Pons, J. L.; Dwinell, M. R.; Hayman, G. T.; Kaldunski, M. L.; Kwitek, A. E.; Laulederkind, S. J. F.; Tutaj, M. A.; Vedi, M.; Wang, S.-J.; D'Eustachio, P.; Aimo, L.; Axelsen, K.; Bridge, A.; Hyka-Nouspikel, N.; Morgat, A.; Aleksander, S. A.; Cherry, J. M.; Engel, S. R.; Karra, K.; Miyasato, S. R.; Nash, R. S.; Skrzypek, M. S.; Weng, S.; Wong, E. D.; Bakker, E.; Berardini, T. Z.; Reiser, L.; Auchincloss, A.; Axelsen, K.; Argoud-Puy, G.; Blatter, M.-C.; Boutet, E.; Breuza, L.; Bridge, A.; Casals-Casas, C.; Coudert, E.; Estreicher, A.; Livia Famiglietti, M.; Feuermann, M.; Gos, A.; Gruaz-Gumowski, N.; Hulo, C.; Hyka-Nouspikel, N.; Jungo, F.; Le Mercier, P.; Lieberherr, D.; Masson, P.; Morgat, A.; Pedruzzi, I.; Pourcel, L.; Poux, S.; Rivoire, C.; Sundaram, S.; Bateman, A.; Bowler-Barnett, E.; Bye-A-Jee, H.; Denny, P.; Ignatchenko, A.; Ishtiaq, R.; Lock, A.; Lussi, Y.; Magrane, M.; Martin, M. J.; Orchard, S.; Raposo, P.; Speretta, E.; Tyagi, N.; Warner, K.; Zaru, R.; Diehl, A. D.; Lee, R.; Chan, J.; Diamantakis, S.; Raciti, D.; Zarowiecki, M.; Fisher, M.; James-Zorn, C.; Ponferrada, V.; Zorn, A.; Ramachandran, S.; Ruzicka, L.; Westerfield, M. The Gene Ontology Knowledgebase in 2023. *Genetics* **2023**, *224* (1), iyad031. https://doi.org/10.1093/genetics/iyad031.

(9) Paysan-Lafosse, T.; Blum, M.; Chuguransky, S.; Grego, T.; Pinto, B. L.; Salazar, G. A.; Bileschi, M. L.; Bork, P.; Bridge, A.; Colwell, L.; Gough, J.; Haft, D. H.; Letunić, I.; Marchler-Bauer, A.; Mi, H.; Natale, D. A.; Orengo, C. A.; Pandurangan, A. P.; Rivoire, C.; Sigrist, C. J. A.; Sillitoe, I.; Thanki, N.; Thomas, P. D.; Tosatto, S. C. E.; Wu, C. H.; Bateman, A. InterPro in 2022. *Nucleic Acids Res.* **2023**, *51* (D1), D418–D427. https://doi.org/10.1093/nar/gkac993.

(10) Van Den Bossche, T.; Kunath, B. J.; Schallert, K.; Schäpe, S. S.; Abraham, P. E.; Armengaud, J.; Arntzen, M. Ø.; Bassignani, A.; Benndorf, D.; Fuchs, S.; Giannone, R. J.; Griffin, T. J.; Hagen, L. H.; Halder, R.; Henry, C.; Hettich, R. L.; Heyer, R.; Jagtap, P.; Jehmlich, N.; Jensen, M.; Juste, C.; Kleiner, M.; Langella, O.; Lehmann, T.; Leith, E.; May, P.; Mesuere, B.; Miotello, G.; Peters, S. L.; Pible, O.; Queiros, P. T.; Reichl, U.; Renard, B. Y.; Schiebenhoefer, H.; Sczyrba, A.; Tanca, A.; Trappe, K.; Trezzi, J.-P.; Uzzau, S.; Verschaffelt, P.; von Bergen, M.; Wilmes, P.; Wolf, M.; Martens, L.; Muth, T. Critical Assessment of MetaProteome Investigation (CAMPI): A Multi-Laboratory Comparison of Established Workflows. *Nat. Commun.* **2021**, *12* (1), 7305. https://doi.org/10.1038/s41467-021-27542-8.

(11) Olsen, J. V.; Ong, S.-E.; Mann, M. Trypsin Cleaves Exclusively C-Terminal to Arginine and Lysine Residues *. *Mol. Cell. Proteomics* **2004**, *3* (6), 608–614. https://doi.org/10.1074/mcp.T400003-MCP200.

(12) Yan, Z.; He, F.; Xiao, F.; He, H.; Li, D.; Cong, L.; Lin, L.; Zhu, H.; Wu, Y.; Yan, R.; Li, X.; Shan, H. A Semi-Tryptic Peptide Centric Metaproteomic Mining Approach and Its Potential Utility in Capturing Signatures of Gut Microbial Proteolysis. *Microbiome* **2021**, *9* (1), 12. https://doi.org/10.1186/s40168-020-00967-x.

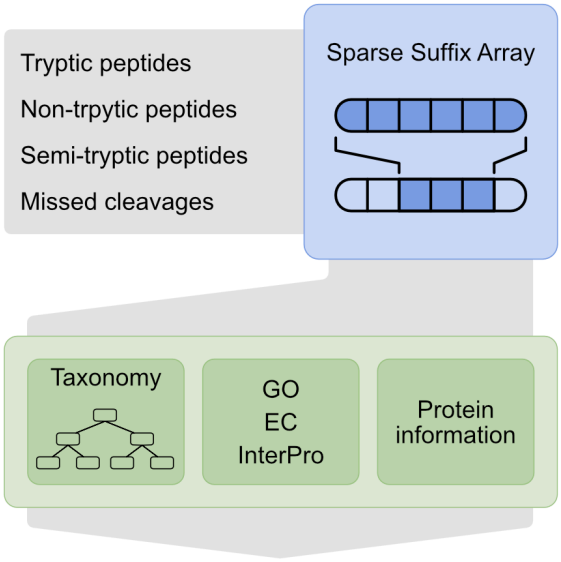
(13) Shapiro, I. E.; Bassani-Sternberg, M. The Impact of Immunopeptidomics: From Basic Research to Clinical Implementation. *Semin. Immunol.* **2023**, *66*, 101727. https://doi.org/10.1016/j.smim.2023.101727.

(14) Mori, Y. Y-256/Libdivsufsort, 2024. https://github.com/y-256/libdivsufsort (accessed 2024-09-17).

(15) Grebnov, I. IlyaGrebnov/Libsais, 2024. https://github.com/IlyaGrebnov/libsais (accessed 2024-09-17).

(16) Mesuere, B.; Devreese, B.; Debyser, G.; Aerts, M.; Vandamme, P.; Dawyndt, P. Unipept: Tryptic Peptide-Based Biodiversity Analysis of Metaproteome Samples. *J. Proteome Res.*

2012, *11* (12), 5773–5780. https://doi.org/10.1021/pr300576s.
(17) Van Den Bossche, T.; Kunath, B. J.; Schallert, K.; Schäpe, S. S.; Abraham, P. E.; Armengaud, J.; Arntzen, M. Ø.; Bassignani, A.; Benndorf, D.; Fuchs, S.; Giannone, R. J.; Griffin, T. J.; Hagen, L. H.; Halder, R.; Henry, C.; Hettich, R. L.; Heyer, R.; Jagtap, P.; Jehmlich, N.; Jensen, M.; Juste, C.; Kleiner, M.; Langella, O.; Lehmann, T.; Leith, E.; May, P.; Mesuere, B.; Miotello, G.; Peters, S. L.; Pible, O.; Queiros, P. T.; Reichl, U.; Renard, B. Y.; Schiebenhoefer, H.; Sczyrba, A.; Tanca, A.; Trappe, K.; Trezzi, J.-P.; Uzzau, S.; Verschaffelt, P.; von Bergen, M.; Wilmes, P.; Wolf, M.; Martens, L.; Muth, T. Critical Assessment of MetaProteome Investigation (CAMPI): A Multi-Laboratory Comparison of Established Workflows. *Nat. Commun.* **2021**, *12* (1), 7305. https://doi.org/10.1038/s41467-021-27542-8.
(18) *Peptide search | UniProt*. https://www.uniprot.org/peptide-search (accessed 2024-09-24).
(19) Chen, C.; Li, Z.; Huang, H.; Suzek, B. E.; Wu, C. H.; UniProt Consortium. A Fast Peptide Match Service for UniProt Knowledgebase. *Bioinforma. Oxf. Engl.* **2013**, *29* (21), 2808–2809. https://doi.org/10.1093/bioinformatics/btt484.
(20) *ScanProsite*. https://prosite.expasy.org/scanprosite/ (accessed 2024-09-24).


Tryptic peptides
Non-trpytic peptides
Semi-tryptic peptides
Missed cleavages
Sparse Suffix Array
Taxonomy
GO
EC
InterPro
Protein information

For TOC Only